

1st Preliminary Design Document for AlumniBook, Team 3

• 系統介紹

1. 系統概述

我們的期末 **project** 開發的是一個促進校友交流的平台。每位校友都有一個 **default** 的帳號與密碼(就如同現行計中的 **email** 帳號一樣，密碼可以再更改)。

在這個平台中大致分成兩個部分，一個是個人的頁面，另一個是論壇。個人頁面中會放上自己的資訊，例如：姓名、級別、職業.....等等，當然也可以瀏覽別人的頁面。而在論壇中，每個人都可以發表 **issue** 或是參與其他 **issue** 的討論。在後面的文件中會有詳細的功能介紹。

此平台最主要的目的是增加校友的互動，互相提供職業或是產業的資訊交流，也可以透過這個平台舉辦活動增進校友之間的感情，而我們還有一個任務就是增加校友使用平台的頻率，讓平台上的資訊源源不絕。

2. 使用技術與語言

以下是我們平台開發所使用的程式語言與套件：

- 後端語言：Node.js v.10.32
- framework：Express.js v.4 以上
- 前端語言：Angular.js v1.2.*
- DB：MySQL
- 語系套件：i18n

3. 進度

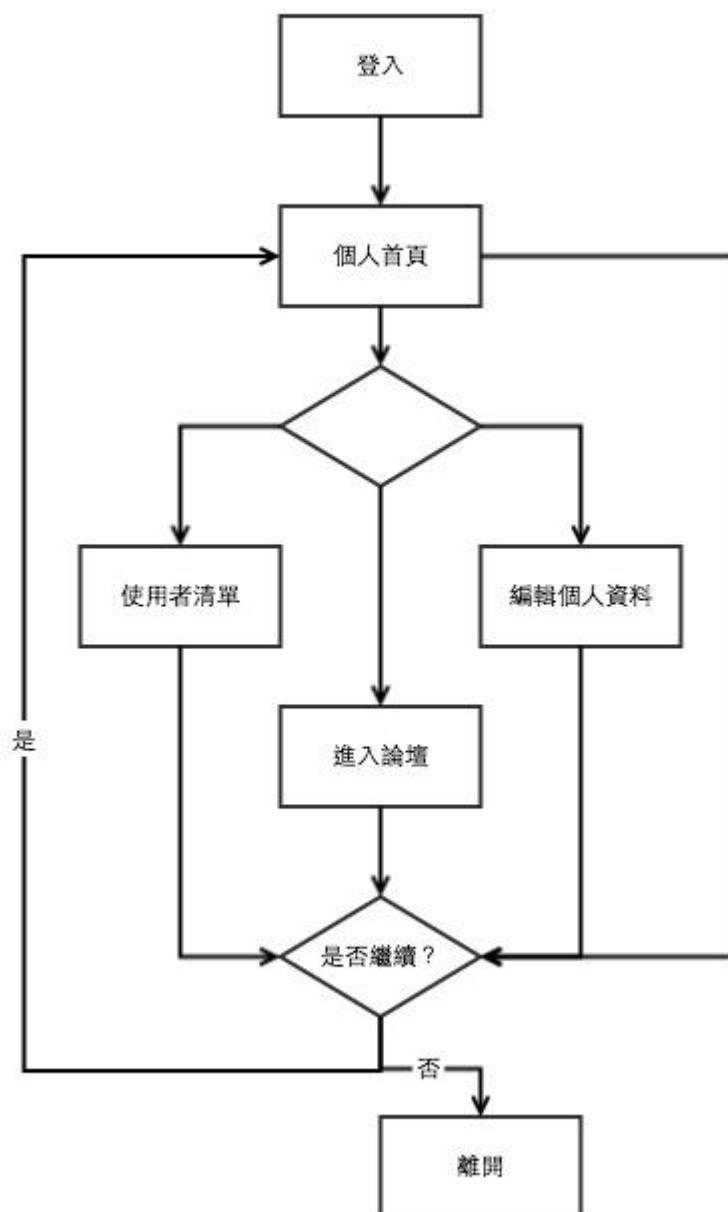
在 **stage 1** 的程式部份我們會完成使用者個人頁面、論壇以及多語言介面，同時也會完成個人編輯頁面以及論壇 **PO** 文頁面，並在 **demo** 時展示我們的 **prototype**。

• 各子元件的設計圖（流程圖、結構圖、UML）

以下我們依照 UML 的規範，分別以活動圖（Activity Diagram）敘述整體系統流程、類別圖（Class Diagram）敘述物件結構。

1. 功能模型（Functional Model）－活動圖（Activity Diagram）

1. 使用者的行為活動圖表：



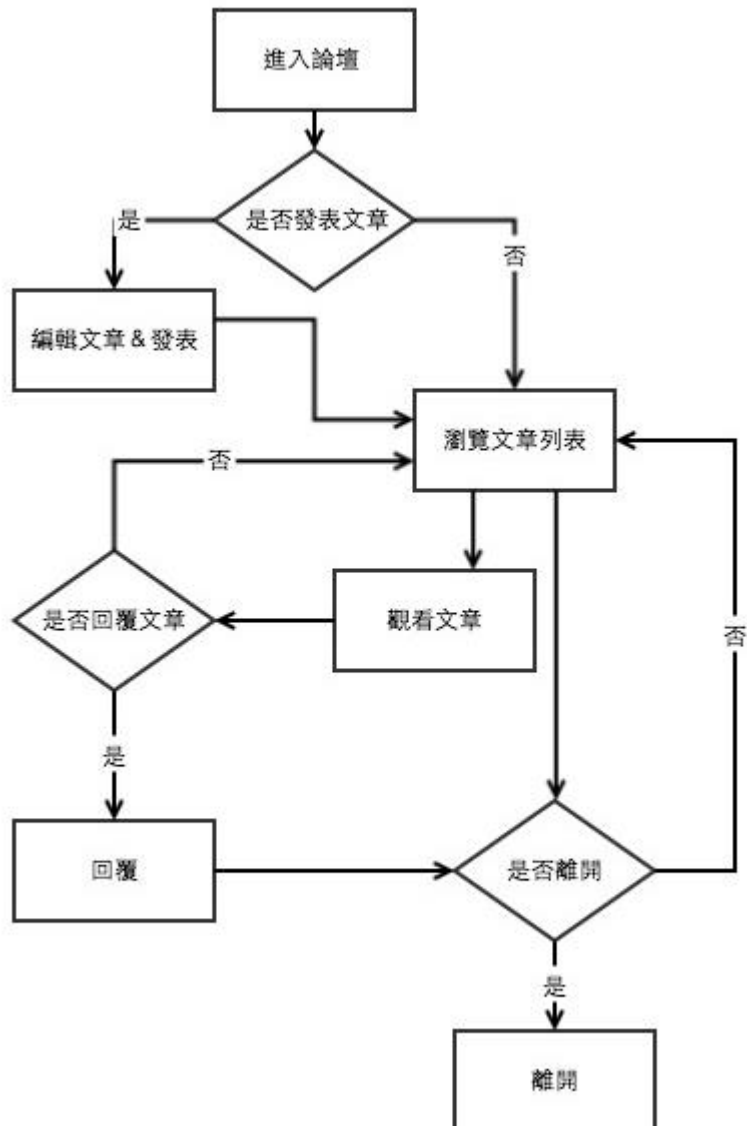
1. 流程介紹：

使用者透過 URL 連進網站後，首先頁面顯示欄位要求使用者輸入帳號及密碼進行登入，經系統判別後若為認可的帳號，該使用者成功登入系統，反之則要求進行註冊。登入後將頁面導入使用者的個人首頁，接著依照使用者的點選，可進行以下三種選擇：

其一，當使用者點選編輯個人資料，將由個人首頁跳轉至編輯資料頁面，提供使用者進行個人資料更改。此頁面其下又有兩個分支，若選擇儲存，將更改資料儲存至資料庫；反之，若選擇取消則資料庫內容將不會有任何更動，兩分支最後都將導回個人首頁。

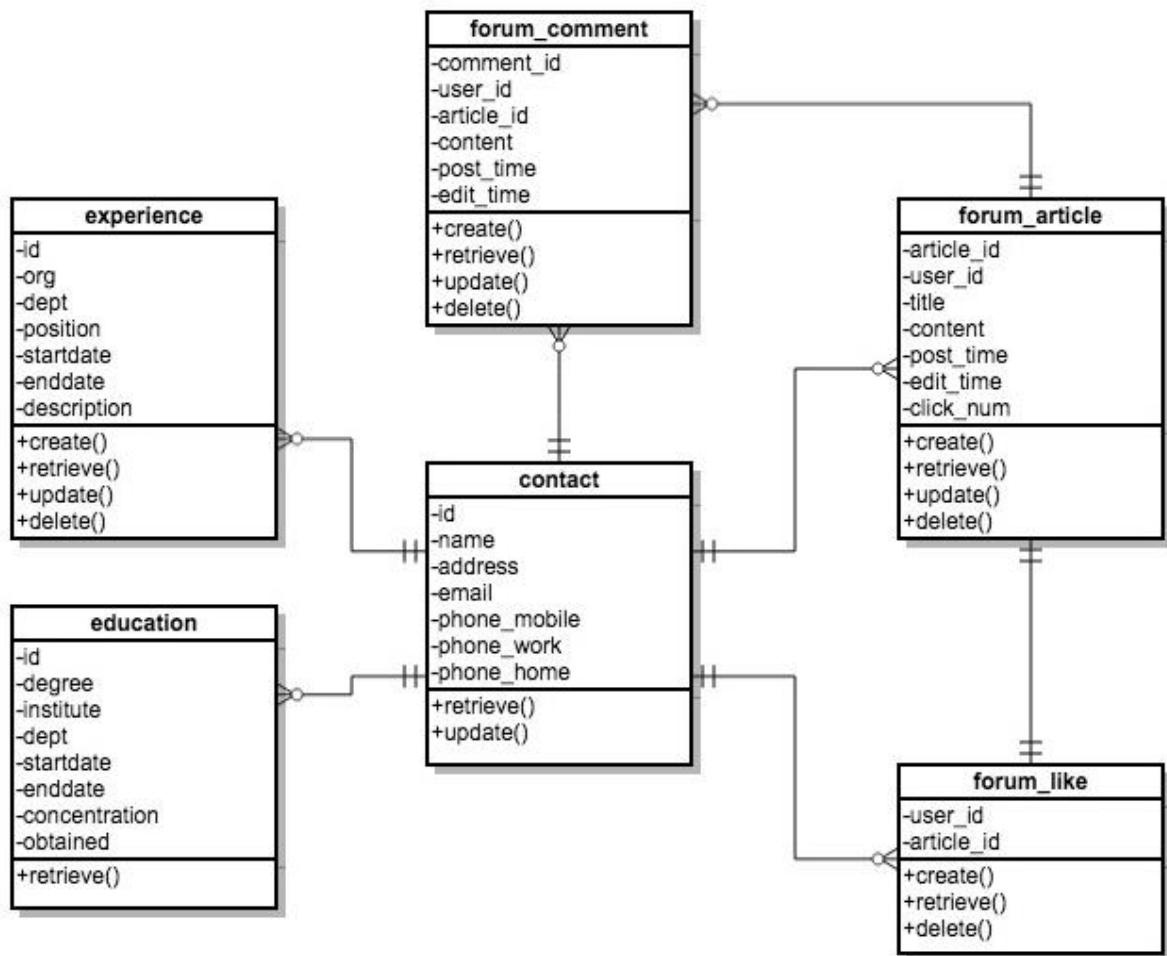
其二則是轉至顯示使用者清單頁，可觀看會員系統中所有使用者的清單列表並進行查詢。

最後則是進入論壇針對主題文章發表議題或給予回應，以上三支分流最後皆可登出離開網站，或是回到個人頁面。



進入論壇後，使用者面臨是否要發表文章的抉擇，是的話則編輯文章並發表，發表後回到文章列表，選擇否則瀏覽文章列表。進入瀏覽文章列表頁之後，使用者決定要觀看文章還是離開系統，若決定觀看文章，則可以回覆文章，若選擇不回覆則將導回文章列表頁面，依使用者選擇端看是否離開論壇。

2. 結構模型 (Structural Model) — 類別圖 (Class Diagram)



- **contact** : id、名字、地址、email、行動電話、辦公室電話、家裡電話，可以修改自己的資料，查詢自己或別人的資料。
- **education** : id、教育程度、institute、系所、就讀時間、concentration、obtained，可以查詢自己或別人的資料。
- **experience** : id、公司、部門、職位、任職時間、描述，可以新增、修改、刪除自己的資料，查詢自己或別人的資料。
- **forum_article** : article_id、user_id、標題、內容、發表時間、修改時間、閱讀數量，可以新增、修改、刪除自己的資料，查詢自己或別人的資料。

- `forum_comment` : `comment_id`、`user_id`、`article_id`、內容、發表時間、修改時間，可以新增、修改、刪除自己的資料，查詢自己或別人的資料。
- `forum_like` : `user_id`、`article_id`，可以新增、刪除自己的資料，查詢自己或別人的資料。

• 子元件介紹（詳細解釋其功能）

針對以下幾個主要頁面，將進行初步的子元件介紹，

1. 登入頁面：

登入頁面大致上擁有三個元件：帳號輸入欄位、密碼輸入欄位，以及送出鍵。一旦送出鍵被按下，頁面將前兩欄位的輸入值帶至後端伺服器，由後端進行身分認證，並從資料庫中取得使用者相關資料，同時導向個人首頁並將資料顯示於上面。

2. 個人首頁：

個人首頁所擁有的元件如下：登出鍵、編輯鍵。前者顧名思義，按下之後將觸發登出事件，該名使用者的登入狀態遭後端伺服器釋放，接著頁面跳轉回登入首頁。至於使用者按下編輯鍵後，將使得網頁後端伺服器從資料庫取出該名使用者的相關資料，同時導向編輯頁面，並將該份資料顯示於可編輯欄位內做為預設值，此預設值可供使用者進行編輯。

3. 個人資料編輯頁：

個人資料編輯頁具有數個輸入欄位，如文字區塊、下拉式選單、按鈕選單...等，其預設值為與該使用者的對應基本資料以供更改。此外，該頁面尚具有三個按鈕，登出鍵、返回鍵、儲存鍵。登出鍵的觸發事件同前述；返回鍵則將頁面轉回個人首頁，資料庫內容不做任何異動；儲存鍵將所有輸入欄位內容一併傳送至後端伺服器，接著儲存在資料庫內對應位置後，指示前端頁面將轉回到個人首頁。

4. 論壇頁面：

使用者進入論壇頁面後，後端伺服器依照使用者身分，以及其符合該身分的文章閱讀權限，回傳數篇文章標題顯示於頁面。因此，此頁面主要供使用者概略瀏覽文章列表。而該頁面上有三個按鈕可供選擇。其一為閱讀文章，使用者能透過點選該按鈕來向伺服器取得該篇文章的詳細內容（包含作者、時間、完整內文、回應...），由瀏覽頁進入該篇文章專頁。

其次為發表文章，按下此按鈕能夠進入一編輯頁面(此頁面介紹詳如後段)。最後則為登出按鈕，功能及觸發相關事件同前段所述。此外，各篇文章當中都有回應該篇文章之功能，當使用者在回應欄位內輸入回覆內容，並按下回覆。回覆按鍵觸發以下幾個事件：將該內容傳送至後端伺服器，再由伺服器儲存至資料庫內，同時新的回應內容會立即顯示在文章下，沒有任何換頁的事件。

5. 文章編輯頁面：

此頁面主要供使用者發表文章之用途，該頁面內含輸入欄位（如：標題填寫、內文區...等）。使用者將選擇取消返回論壇頁面，或是點選儲存按鍵，將上述輸入元件的內容傳送至後端伺服器，並由後端伺服器儲存在資料庫中相對應的欄位。待資料庫中的資料成功存取後，後端伺服器再重新導回論壇頁面。

• 討論課程上的應用

在 stage 1，已經使用 UML 與 Domain Model 做部分的系統建模，利用 Git 版本控制系統做開發，運用 MySQL、PHP、jQuery 等語言及套件於前後端開發上，並使用 Slack (<https://slack.com/>) 做為團隊內部資源共享與溝通系統。